3|PRts

NETWORK MANAGEMENT SYSTEM

The present invention belongs to the field of communication systems, especially of optical communication networks, more particularly, to dense wavelength division multiplexed optical networks with arbitrary topology, e.g., point-two-point, ring, mesh, etc.

The soaring demand for virtual private networks, storage area networking, and other new high speed services are driving bandwidth requirements that test the limits of today's optical communications systems. In an optical network, a node is physically linked to another using one or more optical fibres (cf. Fig. 1). Each of the fibres can carry as many as one hundred or more communication channels, i.e., wavelengths in WDM (Wavelength Division Multiplex) or Dense WDM (DWDM) systems. Thus, for example, for a node with three neighbours as many as three hundred or more wavelength signals originate or terminate or pass through a given node. Each of the wavelengths may carry signals with data rates up to 10 Gbit/s or even higher. Thus each fibre is carrying several terabits of information. This is a tremendous amount of bandwidth and information that must be managed automatically, reliably, rapidly, and efficiently. It is evident that large amount of bandwidth needs to be provisioned. Fast and automatic provisioning enables network bandwidth to be managed on demand in a flexible, dynamic, and efficient manner. Another very important feature of such DWDM networks is reliability or survivability in presence of a failure such as an inadvertent fibre-cut, various types of hardware and software faults, etc. In such networks, in case of a failure, the user data is automatically rerouted to its destination via an alternate or restoration path.

In general, such networks are managed by a network management system which is adapted especially for a single existing network. However, when the existing network, especially its topology, is changed, the network management system must be re-configured by manually adapting of the hardware and software of several nodes. This is an expensive and time-consuming work, especially in the case of meshed networks. Furthermore, the known network management systems are not able to be implemented in networks with an arbitrary topology without manual adaptation of the network management system.

It is an object of the present invention to overcome the disadvantages of the state of the art and especially to provide a network management system that could be implemented in a network with an arbitrary topology, and which provides a highly flexible and reliable managing of the network.

The object of the invention is realized by a method according to claim 1 and a network management system according to claim 11. The sub-claims provide preferable embodiments of the present invention.

In the network, especially the optical network, which is managed by the method according to the present invention multiple nodes are interconnected in an arbitrary topology. The management system is able to manage the whole network and provides intelligence for efficient and optimal use of network resources. The management system comprises preferably various software modules in each node. One software module is a node manager, for example, which takes care of the network management activities. A node manager in each node communicates with other node managers in other nodes through the supervisory network. The supervisory network is formed with the help of supervisory channels between the various nodes of the network. A physical supervisory channel between two nodes in the network might be carried over optical fibre or other types of transport media. Node managers in different nodes might communicate over logical supervisory connections established over one or more physical supervisory channels between various nodes. These logical supervisory connections might be configured manually or with the help of software modules in one or more network nodes. In a preferred embodiment this is done by using a software module called NetProc, which is described in the application PCT/EP03/02704, which has been filed on 14.03.2003 by the same applicant, and which is incorporated by reference into the present application. The NetProc provides the following supervisory network features:

1) Supervisory connection establishment between two network nodes. Each node can have one or more NetProcs. This architecture allows establishment of a direct logical supervisory connection between any arbitrary pair of nodes interconnected by the supervisory channel. Fault-tolerant or redundant connections through two or more paths. In a preferred embodiment these paths are node and link disjoint, as will be described in more detail. The management system uses NetProc's services to exchange messages with other nodes. Any supervisory data is sent through one or several or all of the available redundant connections. Each message is given a sequence number. On the receiving end the duplicate messages are discarded and only one, for example the first, of the arriving message is passed on to the supervisory management layer.
2) Hardware fault and software error detection on all paths of the supervisory channel and the associated auto-recovery to re-establish the supervisory channel. Error checking in the data transmission is done by using sequence numbers on the messages. The status of each connection is monitored by sending keep-alive messages at regular intervals. In the event that a reply to keep-alive message is not received within a specified time the connection is explicitly closed and the two nodes try to re-establish connection between themselves. The closing of connection(s) and attempts to re-establish them are done automatically.
3) Relaying information reliably to one or more network managers running on one or more network nodes or other work stations.
4) The management of the network is carried out by a node manager present in each node or at one or more nodes or other centralized locations. The various node managers communicate using the NetProc.

A preferred supervisory network has the flexibility to be configured by standard protocols like OSPF, MPLS or by using NetProc. Following features apply:

- The supervisory network topology is automatically discovered with the help of OSPF. Each node manager executes a single OSPF and the OSPF in each node is configured to talk with neighbouring nodes.
- The nodes discover their neighbours and exchange Link State Advertisements. Once the Link State adjacencies are formed and the OSPF converges on the topology, each node possesses the routing table and is able to reach other nodes over the supervisory channel.
- The status of the supervisory channel is monitored by OSPF and in the event of link failure the alternate routes are configured. Fault-tolerant connections are set up using two or more Label Switched Paths over two or more disjoint paths to each destination. Thus a signalling message sent to a node travels through multiple Label Switched Paths and reaches its appropriate destination.

According to the present invention a node module is provided in each node manager. Thereby, the module could be implemented in form of software or hardware or both. The node module in each node provides an interface to the hardware of the corresponding node. By each node module the hardware settings of the respective node could be amended and/or monitored.

At least one node manager is provided with a master module. The master module could also be implemented in form of software and/or hardware. The master module communicates through supervisory connections with the various node modules and controls the various amendments carried out by the different node modules and/or processes the hardware settings of the different nodes monitored by the corresponding node modules.

Preferably, not only one but several or all of the node managers in the different nodes comprise a master module. Preferably, in this case the master module has an active state and a passive state which the master module might be set to. Further preferably, at a given time only one master module is allowed to be set to the active state. Such a master module might be called the Master and all the other master modules, which are in a passive state, might be called Deputy Master (DM). Only the master module that is in the active state (Master) controls the different amendments of hardware settings carried out by the node modules and processes the hardware settings monitored by the node modules.

Preferable embodiments of the present invention will be described in the following with reference to the accompanying drawings, in which

Figure 1 shows a preferable first architecture of a node manager;

Figure 2 shows the established supervisory connections between corresponding different nodes;

Figure 3 shows a second preferable architecture of a node manager with an attached master controller;

Figures 4 and 5 show reduced supervisory connections used in the shown second architecture.

The functions of a node manager (1) according to the embodiment shown in Figure 1 are separated into two main modules. The node module (2) takes care of the activities local to a node. Every node has a node module (2), which connects to one or more master modules (3) located at the same node or other nodes using the supervisory channel. Among other things, the node module (2) provides interface to the hardware and allows the master module (3) to make any changes or informs the master module (3) of any changes in the hardware properties. The second module called master module (3) is present in one or several or all nodes. The master module (3) includes MasterProc (5) for global and local network management, DBProc for database related tasks and features, Interface to GUI (4) to support the hardware element management and local and global network management. This is shown in Fig 1. Thereby, the term "Proc" denotes one or more software modules with predetermined functionality.

In addition to the node manager (1), there is a Graphical User Interface (GUI), which is used to input (or enter), output (or view), and modify various parameters and/or properties related to the node hardware. The GUI is also used to input (or enter), output (or view), and modify various parameters and/or properties related to the local and/or global network management. The GUI is connected to the master module (3) (cf. Fig. 1).

The functions of a master module (3) include

- Receiving/sending node information from/to one or more nodes, reading, writing, and updating the database (DB) and providing an interface to the GUI.
- Accepting user and/or hardware commands for modifying and/or updating node properties and sending them to the relevant nodes. Such commands may also be received from other nodes.
- Processing network management related commands and messages, e.g., demand information from the user, which includes creation of demand, selection of one or more demand-paths, starting and stopping traffic for a demand, etc..
- Monitoring the status of demands and providing protection or restoration actions in the event of one or more faults and/or errors in a demand.
- Exchange of heartbeat messages and related processing
- Database synchronization

The master module (3) according to the shown embodiment provides the following interfaces

- Interface to the node module (2) in one or several or all nodes

- Interface to the database
- Interface to the GUI (4) in one or several or all nodes

Although there are several master modules (3) located in several network nodes, at a given time only one master module (3) may be active. Such a master module (3) might be designated as the Master and all the other master modules (3) as a Deputy Master (DM). Further, a master module (3) performs the tasks of the Master or a Deputy Master depending on the configuration. Such a configuration can be done statically or dynamically. It may also be done manually or automatically.

The node module (2) in each node needs a connection to the master module (3) and vice-versa. This connection is set-up over the supervisory channel using NetProc or equivalent software modules.

The Master located in a particular node coordinates all the network management activities. The Master is an essential part of the network management and needs to be functional all the time. It therefore becomes important to make sure that there is a backup or standby module, which takes over when the Master fails for some reason. For this purpose one or more Deputy Masters are designated as the backup or standby to the Master. These Deputy Masters take over the functions of the master module (3) when the Master fails. The master module (3) has different functionality based on whether it is the Master or a Deputy Master. The nodes where the Master and a Deputy Master are located are termed as the master node and a DM node, respectively. Finally, a full set of supervisory connections between all pairs of nodes which contain master module (3) are required in order to manage the redundancy and fault-tolerance with respect to the Master functionality. A full set of supervisory connections implies a supervisory connection between all pair of nodes. A reduced set of supervisory connections is defined as a set of those connections between a pair of nodes in which one of the nodes is the master node.

As the node manager software first comes up, a node preferably is always initialised to be a Deputy Master node. Following protocol is used in determining as to which node acts as a Master at a given time: 1) All nodes periodically exchange Heartbeat messages among each other, the contents of which are used to determine as to which node is the master node and also to monitor the status of master node by the various Deputy Master nodes. 2) A Heartbeat message contains the node ID of the sender node as well as its status, either Master or Deputy-Master. 3) The receiving node first examines the status of all the received Heartbeat Messages within a certain time interval. If it receives a Master status in any of the received Heartbeat messages, it remains in the same state as before without altering its status. If it does not receive a Master status in any of its Heartbeat messages, it compares its ID with other received IDs. If its ID is smaller than the received IDs it assumes the role of Master otherwise it remains in the same state as before without altering its status. As an alternative, if on start-up a node does not receive Heartbeat message from other nodes after sending a configurable number of Heartbeat messages it assumes the role of the Master. 4) If and only if the existing Master fails the new Master election process takes place. Master election is done by processing heartbeat messages as discussed above. 5) In case two nodes assume for any reason the unintended role of a master node it is resolved using the following protocol. Among the different master nodes the node with the lowest ID number retains the role of Master, all other master nodes revert their role to being a Deputy Master node.

Based on the contents of heartbeat messages there may be other procedures for selecting as to which master module acts as the Master, for example the master module in the node with the largest ID.

After the election is over, the master module (3) in master node takes over the operations of the network and performs the network management functions. The change of role of a particular node from a Deputy Master node to a master node should be performed as quickly and as seamlessly as possible to have minimum disruption in network operation. The master node and Deputy Master nodes perform additional functions for fault-tolerance. These include among other functions database synchronization between master node and Deputy Master nodes.

In the following sections two architectures for handling redundancy and fault-tolerance are presented.

The node manager corresponding to a first architecture is shown in Fig. 1. The master node and all the Deputy Master nodes are connected through the supervisory channel configured by NetProc or an equivalent software module. Using such supervisory connections (10) between each pair of nodes, each node module in each node sends all node-related information to the master node and to all the Deputy Master nodes as shown in Fig. 2, e.g., for a four node network. Exchange of heartbeat messages and related processing is done as discussed previously in this document.

The database in the master node and a Deputy Master node needs to be synchronized at all times. This ensures correct operation when the master node fails and a new master node is elected. After a new master node is elected, it sends the current dump (state) of the database to all other Deputy Master nodes before resuming its duty as a master node. This makes sure that the database in all nodes are synchronized before the nodes begin their management function. During normal operation, both the master node and all Deputy Master nodes receive messages from node modules in all nodes. Thus, the master module in each node updates the database located in that particular node. The difference in the functionality of Master versus Deputy Master is that a node acting as Deputy Master does not send any message to other nodes but only receives all node-related messages. The primary function of a Deputy Master node in this architecture is to perform the database synchronization. When a node comes up again after a failure and a master node already exists then the restored node requests for the current dump of the database from the master node.

In the second architecture, there is an additional software module running at a node, namely, master controller as shown in Fig 3. The so-called master controller (4) is a module, which could be implemented by software and/or hardware.

The Node Module (2) and master controller (7) are active in all nodes of the network. However, the master module (3) is active only in the master node. In this architecture, it is the master controller (7) which takes part in master-election and role-change related steps, e.g., database synchronization. When the nodes come up for the first time, the Node Module (2) and master controller (7) are started in each node. The master module (3) is not started initially. The master controllers (7) in various nodes by exchanging and processing heartbeat messages among each other elect a particular node as the master node. Thereafter, it starts the master module (3) only in the master node (cf. Fig 3).

The master controller (7) in each node is connected to all other master controllers (7) in other nodes through the supervisory channel. The Node Module in different nodes is connected only to the master module (3) as shown in Fig 4 through a reduced set of supervisory connections (10).

When the master node changes, e.g., from node 1 to node 2, the master controller (7) in that node, dynamically and automatically re-configures the connection between the node modules (2) and the new master module (3) as shown in Fig 5.

This dynamic reconfiguration is done using NetProc or other similar software modules and the master controller present in each node. The master controller sends a re-configure message to NetProc in each node, with the node ID of the new master node. The NetProc in each node on receiving the message re-configures the connections so that all the nodes have a logical supervisory connection to the new master node. The nodes can also be statically connected as in architecture 1 and the dynamic reconfiguration step can be avoided.

Exchange of heartbeat messages and related processing is done as discussed previously in this document.

The master controller (7) does the database synchronization between a pair of nodes. After a new master node is elected, the master controller (7) sends the current dump (state) of the database to all the master controllers(7) in Deputy Master nodes before starting the master module processes in the master node. This makes sure that the database in all nodes are synchronized before the nodes begin the management function. The master module (3) informs the master controller (7) of any changes in database and these changes are sent to all other master controllers (7) in other nodes in the network. The master controller (7) in other Deputy Master nodes on receiving the changes from the master node updates the local database. This keeps the database synchronized with the master node. When a node comes up again after a failure and a master node already exists then the restored node requests for the current dump of the database from the master node.